```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

**FILE**ID**LIBFINCVT

```
LL            IIIIII    BBBBBBBB    FFFFFFFFFF    IIIIII    NN      NN    CCCCCCCC  VV      VV  TTTTTTTTTT
LL            IIIIII    BBBBBBBB    FFFFFFFFFF    IIIIII    NN      NN    CCCCCCCC  VV      VV  TTTTTTTTTT
LL              II      BB      BB  FF              II      NN      NN  CC          VV      VV      TT
LL              II      BB      BB  FF              II      NN      NN  CC          VV      VV      TT
LL              II      BB      BB  FF              II      NNNN    NN  CC          VV      VV      TT
LL              II      BB      BB  FF              II      NNNN    NN  CC          VV      VV      TT
LL              II      BBBBBBBB    FFFFFFFF        II      NN  NN  NN  CC          VV      VV      TT
LL              II      BBBBBBBB    FFFFFFFF        II      NN  NN  NN  CC          VV      VV      TT
LL              II      BB      BB  FF              II      NN    NNNN  CC          VV      VV      TT
LL              II      BB      BB  FF              II      NN    NNNN  CC          VV      VV      TT
LL              II      BB      BB  FF              II      NN      NN  CC            VV  VV        TT        ....
LL              II      BB      BB  FF              II      NN      NN  CC            VV  VV        TT        ....
LLLLLLLLLL    IIIIII    BBBBBBBB    FF            IIIIII    NN      NN    CCCCCCCC      VV          TT        ....
LLLLLLLLLL    IIIIII    BBBBBBBB    FF            IIIIII    NN      NN    CCCCCCCC      VV          TT        ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 %TITLE 'LIB$$FIND_CVT_PATH  for internal use of LIB$CVT_DX_DX'
0 MODULE LIB$$FIND_CVT_PATH (                  ! DFA of general data type conversion.
0                    IDENT = '1-006'           ! File:LIBFINCVT.B32 Edit: STAN1006
0                    ) =
1 BEGIN
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:     General Utility Library
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains LIB$$FIND_CVT_PATH routine which is called only
1 !       by LIB$CVT_DX_DX routine.  The reason that these two routines are in
1 !       different modules is because of anticipation of future updates to this
1 !       data conversion routines.  They are very large, and it is easier to
1 !       update them seperately.
1 !
1 ! ENVIRONMENT:  User mode - AST reentrant
1 !
1 ! AUTHOR:       Farokh Morshed          01-09-1981
1 !
1 ! MODIFIED BY:
1 !
1 ! 1-001 - Original.  FM1001      01-09-1981
1 ! 1-002 - Put in a check for DSC$W_LENGTH to be 1 when class A, or NCA, and
1 !         if class NCA stride must be 1.  FM 9-9-81
1 ! 1-003 - Put in a new data type, DSC$K_DTYPE_VT.  FM 1-DEC-81.
1 ! 1-004 - Put in a feature where DST_INFO [D_LEN] can be picked up for
1 !         LIB$CVT_DX_DX. FM 2-DEC-81.
1 ! 1-005 - Fix the bug that in [K_S_NLO, K_SD_NLO] negative inputs are picked
1 !         up as positive.  FM 1-Mar-83
1 ! 1-006 - Remove informational errors. STAN 24-Jul-1984.
1 !--
```

```
;   58          0058  1
```

```
1 %SBTTL 'Declarations'
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !+
1 ! LINKAGE
1 !-
1
1 LINKAGE
1     JSB_R1 = JSB (REGISTER = 0, REGISTER = 1) : PRESERVE (0, 1);
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     LIB$$FIND_CVT_PATH;                              ! Routine to find the conversion
1
1                                                      ! being done and report any
1                                                      ! unsupported fields in the descriptors.
1 !
1 ! INCLUDE FILES:
1 !
1
1 LIBRARY 'RTLSTARLE';                                 ! System symbols, from SYS$LIBRARY:STARLET.L32
1
1 REQUIRE 'RTLIN:RTLPSECT';                            ! Define PSECT declarations macros
1
1 !
1 ! PSECTS:
1 !
1 DECLARE_PSECTS (LIB);                                ! Declare PSECTs for LIB$ facility
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
```

```
1 %SBTTL 'Deterministic Finite Automata for LIB$CVT_DX_DX'
1
1 GLOBAL ROUTINE LIB$$FIND_CVT_PATH (              ! Deterministic Finite Automata
1                                                  ! that will parse the source
1                                                  ! and destination descriptors.
1         SOURCE                                   ! Source descriptor that was passed
1                                                  ! to LIB$CVT_DX_DX.
1     , DESTINATION                                ! Destination descriptor that was
1                                                  ! passed to LIB$CVT_DX_DX.
1     , SRC_INFO                                   ! Address of a record that this
1                                                  ! routine will put the source
1                                                  ! information in.
1     , DST_INFO                                   ! Address of a record that this
1                                                  ! routine will put the destination
1                                                  ! information in.
1     , CVT_PATH                                   ! This code will determine what
1                                                  ! label of the LIB$CVT_DX_DX
1                                                  ! routine's CASE statement will
1                                                  ! be taken.
1     ) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine is comprised of a Deterministic Finite Automaton, defined
1 !       as a 5 tuple :
1 !       STATES          : There is a state for each CLASS, and CLASS, DATA TYPE
1 !                         combination.
1 !       Alphabet        : Classes and Data types.
1 !       Mappings        : M(CLASS_S , DTYPE_B) := CLASS_S_DTYPE_B
1 !                         . . . . . . . . . . . . . . . . . . .
1 !                         . . . . . . . . . . . . . . . . . . .
1 !                         M(CLASS_D , DTYPE_W) := error
1 !                         . . . . . . . . . . . . . . . . . . .
1 !                         . . . . . . . . . . . . . . . . . . .
1 !       Start state     :
1 !       Final states    : All possible combinations of CLASS, DTYPE.
1 !                         Some of these combinations are allowed, others
1 !                         are not. The error combinations are denoted by
1 !                         negative numbers as states.
1 !
1 ! MAINTENANCE OF THIS ROUTINE :
1 !
1 !This routine knows about all classes and data types of Appendix C V8.3.
1 !(You may want to update the above line everytime a change is made)
1 !To make an already existing CLASS, DATA TYPE combination a valid one, as
1 !opposed to an error you must :
1 !       1. Insert the symbol for that data type in DTYPE_TABLE in place of the
1 !          error state.
1 !       2. Define a FINAL_STATE for this combination.
1 !       3. Give it an action routine.
1 !
1 !To add a new data type you must :
1 !       1. Increment K_MAX_DATA_TYPES.
1 !       2. Set K_MAX_DTYPE_STA to value of the new data type.
1 !       3. Does any of the following need to be changed ?
1 !               a. K_SMLFINSTA
```

```
1 !                             b. K_LRGFINSTA
1 !                             c. K_TOP_SD
1 !                             d. K_BOTTOM_SD
1 !                     4. Define a new FINAL_STATE.
1 !                     5. Each category in DTYPE_TABLE must have a new entry for the data type.
1 !                        Note that the position (starting at 0) of each entry in each category is equivalent
1 !                        to the data type value.
1 !                     6. Add the new lable into the action routines CASE statement and
1 !                        the sub-CASE statements in LIB$CVT_DX_DX will need to be modified to
1 !                        include this new data type.
1 !
1 !To add a new class you must :
1 !                     1. Increment K_MAX_CLASSES
1 !                     2. Set K_MAX_CLASS_STA to value of the new class.
1 !                     3. Increment K_ACTUAL_CLASSES.
1 !                     4. Make a new K_STATEx_CLASS_y, where x is class value and y is the
1 !                        symbol of the class.
1 !                     5. Make a new FINAL_STATE.
1 !                     6. Add a new category to the STATES structure at the end, with a index
1 !                        value of one higher than the last category.
1 !                     7. Make a new entry in CLASS_TABLE.
1 !                     8. Make a new category in DTYPE_TABLE.
1 !                     9. Make a new lable in the action routine CASE statement.
1 !
1 !
1 ! CALLING SEQUENCE:
1 !
1 !                     ret_status.wlc.v = FIND_CVT_PATH ( SOURCE.rx.dx,
1 !                                                        DESTINATION.rx.dx,
1 !                                                        SRC_INFO.wr.r,
1 !                                                        DST_INFO.wr.r,
1 !                                                        CVT_PATH.wlu.r )
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !                     SOURCE          Address of source descriptor passed to LIB$CVT_DX_DX.
1 !                     DESTINATION     Address of destination descriptor passed to LIB$CVT_DX_DX.
1 !                     SRC_INFO        Address of a record in LIB$CVT_DX_DX.
1 !                     DST_INFO        Address of a record in LIB$CVT_DX_DX.
1 !                     CVT_PATH        Address of a longword in LIB$CVT_DX_DX.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !                     NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !                     NONE
1 !
1 ! COMPLETION STATUS: (or ROUTINE VALUE:)
1 !
1 !                     K_UNSCLAROU             : -1 Unsupported CLASS by routine.
1 !                     K_UNSDTYROU             : -2 Unsupported DTYPE by routine.
1 !                     K_UNSDESROU             : -3 Unsupported descriptor by routine.
1 !                     K_UNSDESSTA             : -4 Unsupported descriptor by standard.
1 !                     K_UNSCLASTA             : -5 Unsupported CLASS by standard.
1 !                     K_UNSDTYSTA             : -6 Unsupported DTYPE by standard.
```

```
;          1 !      K_INVNBDS                 : -7 Invalid NBDS because array size is greater
;          1 !                                     than WU or dimension is not one.
;          1 !      K_SUPPORTED               :  1 This descriptor is supported.
;          1 !
;          1 ! SIDE EFFECTS:
;          1 !
;          1 !      Caller of LIB$CVT_DX_DX must have LIB$EMULATE as a handler, if the
;          1 !      source or destination descriptor explicitely ask for G, H, O conversions.
;          1 !
;          1 !--
;          1
;          2    BEGIN
;          2
;          2    BUILTIN
;          2        CVTTP,
;          2        CVTSP,
;          2        CVTPT,
;          2        CVTPS,
;          2        CMPP;
;          2
;          2 !+
;          2 ! MACRO
;          2 !-
;          2 !<BLF/MACRO>
;          2
;          2    MACRO
;          2 !+
;          2 ! These MACROs are used for clarity of code, since there is not builtin for them.
;          2 !-
;        M 2        CVTGH =
;          2            LIB$$CVT_CVTGH_R1 %,
;          2 !+
;          2 ! These MACROs define portions of intermediate data buffer.
;          2 !-
;        M 2        LONG_1 =
;          2            0, 0, 32, 0 %,
;        M 2        LONG_2 =
;          2            4, 0, 32, 0 %,
;        M 2        LONG_3 =
;          2            8, 0, 32, 0 %,
;        M 2        LONG_4 =
;          2            12, 0, 32, 0 %,
;        M 2        LONG_5 =
;          2            16, 0, 32, 0 %,
;        M 2        LONG_6 =
;          2            20, 0, 32, 0 %,
;        M 2        LONG_7 =
;          2            24, 0, 32, 0 %,
;        M 2        LONG_8 =
;          2            28, 0, 32, 0 %,
;        M 2        S_LONG_1 =
;          2            0, 0, 32, 1 %,
;        M 2        S_LONG_2 =
;          2            4, 0, 32, 1 %,
;        M 2        S_BYTE_1 =
;          2            0, 0, 8, 1 %,
;        M 2        BYTE_1 =
;  271            0363  2                0, 0, 8, 0 %,
;  272          M 0364  2           BYTE_2 =
;  273            0365  2                1, 0, 8, 0 %,
;  274          M 0366  2           S_WORD_1 =
;  275            0367  2                0, 0, 16, 1 %,
;  276          M 0368  2           WORD_1 =
;  277            0369  2                0, 0, 16, 0 %,
;  278          M 0370  2           WORD_2 =
;  279            0371  2                2, 0, 16, 0 %,
;  280          M 0372  2           NIBBLE_1 =
;  281            0373  2                0, 0, 4, 0 %,
;  282            0374  2   !+
;  283            0375  2   ! This MACRO calculates final states given the state and the token.
;  284            0376  2   !-
;  285          M 0377  2           FINAL_STATE (CLASS, DATA_TYPE) =
;  286            0378  2                CLASS*K_MAX_DATA_TYPES + DATA_TYPE %,
;  287            0379  2   !+
;  288            0380  2   ! This macro is used for SRC_INFO or DST_INFO scale field.
;  289            0381  2   !-
;  290          M 0382  2           M_SCALE =
;  291            0383  2                0, 0, 8, 1 %,
;  292            0384  2   !+
;  293            0385  2   ! This macro is used for SRC_INFO or DST_INFO length field.
;  294            0386  2   !-
;  295          M 0387  2           M_LEN =
;  296            0388  2                5, 0, 16, 0 %,
;  297            0389  2   !+
;  298            0390  2   ! Define the start state data structure of the DFA.
;  299            0391  2   !-
;  300          M 0392  2           START_STATE =
;  301            0393  2                VECTOR [K_MAX_CLASSES, BYTE, SIGNED] %;
;  302            0394  2
;  303            0395  2   !+
;  304            0396  2   ! EXTERNAL
;  305            0397  2   !-
;  306            0398  2
;  307            0399  2       EXTERNAL ROUTINE
;  308            0400  2           LIB$STOP : NOVALUE,
;  309            0401  2           CVTGH : JSB_R1 NOVALUE;
;  310            0402  2
;  311            0403  2   !+
;  312            0404  2   ! These are the translation tables used when translating from or to packed decimal.
;  313            0405  2   !-
;  314            0406  2
;  315            0407  2       EXTERNAL
;  316            0408  2           LIB$AB_CVTTP_U,
;  317            0409  2           LIB$AB_CVT_O_U,
;  318            0410  2           LIB$AB_CVTTP_O,
;  319            0411  2           LIB$AB_CVT_U_O,
;  320            0412  2           LIB$AB_CVTPT_U,
;  321            0413  2           LIB$AB_CVTPT_O,
;  322            0414  2           LIB$AB_CVTPT_Z,
;  323            0415  2           LIB$AB_CVTTP_Z;
;  324            0416  2
;  325            0417  2       EXTERNAL LITERAL                                   ! Condition value symbols
;  326            0418  2           LIB$_FATERRLIB;                                ! Fatal error in library.
;  327            0419  2
```

```
!+
! FIELD DECLARATIONS
!-

    FIELD
        SRC_INFO_FIELDS =
            SET
            S_SCALE = [0, 0, 8, 1],
            S_POINTER = [1, 0, 32, 0],
            S_LEN = [5, 0, 16, 0],
            S_SIGN = [7, 0, 1, 0]
            TES;

    FIELD
        DST_INFO_FIELDS =
            SET
            D_SCALE = [0, 0, 8, 1],
            D_LEN = [5, 0, 16, 0]
            TES;

!+
! Define some literals.
!-

    LITERAL
!+
!Status returned by FIND_CVT_PATH.
!-
        K_UNSCLAROU = -1,                       ! Unsupported CLASS by routine.
        K_UNSDTYROU = -2,                       ! Unsupported DATA TYPE by routine.
        K_UNSDESROU = -3,                       ! Unsupported descriptor by routine.
        K_UNSDESSTA = -4,                       ! Unsupported descriptor by standard.
        K_UNSCLASTA = -5,                       ! Unsupported CLASS by standard.
        K_UNSDTYSTA = -6,                       ! Unsupported DTYPE by standard
        K_INVNBDS = -7,                         ! Invalid NBDS
                                                ! because either array size is larger
                                                ! than a WU or it is not a one
                                                ! dimensional array.
        K_SUPPORTED = 1,                        ! This descriptor is supported, and valid.
!+
!Some general values :
!-
        K_INTMED_DATA_LENGTH = 32,              ! Intermediate data buffer length
        K_LRGST_WU = 65535,
        K_LRGST_LU = 4294967295,                ! Largest unsigned longword.
        K_LRGST_NEG_L = -2147483648,            ! Largest negative longword.
        K_LRGCLSSUP = DSC$K_CLASS_VS,           ! Largest CLASS supported by routine
        K_SMLCLSSUP = DSC$K_CLASS_S,            ! Smallest CLASS supported by routine
        K_MAX_DATA_TYPES = 38,                  !Total number of DATA TYPES in the standard
        K_MAX_CLASSES = 15,                     !Total number of classes supported.
                                                !Including the error case 0.
        K_MIN_CLASS = DSC$K_CLASS_S,            !Smalles class supported.
        K_MAX_CLASS = DSC$K_CLASS_VS,           !Largest class supported.
        K_MAX_CLASS_STA = DSC$K_CLASS_UBA,      !Max. class number supported by standard.
        K_MAX_DTYPE_STA = DSC$K_DTYPE_VT,       !Max. data type number supported by standard.
        K_ACTUAL_CLASSES = 6,                   !Total classes that are allowed by the STATES table.
```

```
;  385         0477  2          K_MSTNEGERR = -7,                               !Most negative error state
;  386         0478  2          K_SMLFINSTA = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_BU),      !Smallest final state supported.
;  387         0479  2          K_LRGFINSTA = FINAL_STATE (DSC$K_CLASS_VS, DSC$K_DTYPE_VT),     !Largest final state supported.
;  388         0480  2          K_TOP_SD = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_H), !Top state for class SD.
;  389         0481  2          K_BOTTOM_SD = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_B),      !Bottom state for class SD.
;  390         0482  2  !+
;  391         0483  2  !These are the values of the members of K_ACTUAL_CLASSES :
;  392         0484  2  !-
;  393         0485  2          K_STATE1_CLASS_S = DSC$K_CLASS_S,
;  394         0486  2          K_STATE2_CLASS_D = DSC$K_CLASS_D,
;  395         0487  2          K_STATE4_CLASS_A = DSC$K_CLASS_A,
;  396         0488  2          K_STATE9_CLASS_SD = DSC$K_CLASS_SD,
;  397         0489  2          K_STATE10_CLASS_NCA = DSC$K_CLASS_NCA,
;  398         0490  2          K_STATE11_CLASS_VS = DSC$K_CLASS_VS,
;  399         0491  2  !+
;  400         0492  2  !These are the final states that are valid CLASS, DATA TYPE combinations.
;  401         0493  2  !The rest of the final states are error states.
;  402         0494  2  !The first argument to the macro is CLASS, and the second is the DATA TYPE.
;  403         0495  2  !-
;  404         0496  2          K_S_BU = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_BU),
;  405         0497  2          K_S_WU = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_WU),
;  406         0498  2          K_S_LU = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_LU),
;  407         0499  2          K_S_B = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_B),
;  408         0500  2          K_S_W = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_W),
;  409         0501  2          K_S_L = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_L),
;  410         0502  2          K_S_Q = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_Q),
;  411         0503  2          K_S_F = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_F),
;  412         0504  2          K_S_D = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_D),
;  413         0505  2          K_S_T = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_T),
;  414         0506  2          K_S_NU = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_NU),
;  415         0507  2          K_S_NL = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_NL),
;  416         0508  2          K_S_NLO = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_NLO),
;  417         0509  2          K_S_NR = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_NR),
;  418         0510  2          K_S_NRO = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_NRO),
;  419         0511  2          K_S_NZ = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_NZ),
;  420         0512  2          K_S_P = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_P),
;  421         0513  2          K_S_G = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_G),
;  422         0514  2          K_S_H = FINAL_STATE (DSC$K_CLASS_S, DSC$K_DTYPE_H),
;  423         0515  2          K_D_T = FINAL_STATE (DSC$K_CLASS_D, DSC$K_DTYPE_T),
;  424         0516  2          K_A_BU = FINAL_STATE (DSC$K_CLASS_A, DSC$K_DTYPE_BU),
;  425         0517  2          K_A_T = FINAL_STATE (DSC$K_CLASS_A, DSC$K_DTYPE_T),
;  426         0518  2          K_SD_BU = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_BU),
;  427         0519  2          K_SD_WU = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_WU),
;  428         0520  2          K_SD_LU = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_LU),
;  429         0521  2          K_SD_B = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_B),
;  430         0522  2          K_SD_W = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_W),
;  431         0523  2          K_SD_L = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_L),
;  432         0524  2          K_SD_Q = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_Q),
;  433         0525  2          K_SD_F = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_F),
;  434         0526  2          K_SD_D = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_D),
;  435         0527  2          K_SD_G = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_G),
;  436         0528  2          K_SD_H = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_H),
;  437         0529  2          K_SD_T = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_T),
;  438         0530  2          K_SD_NU = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_NU),
;  439         0531  2          K_SD_NL = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_NL),
;  440         0532  2          K_SD_NLO = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_NLO),
;  441         0533  2          K_SD_NR = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_NR),
2               K_SD_NRO = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_NRO),
2               K_SD_NZ = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_NZ),
2               K_SD_P = FINAL_STATE (DSC$K_CLASS_SD, DSC$K_DTYPE_P),
2               K_NCA_BU = FINAL_STATE (DSC$K_CLASS_NCA, DSC$K_DTYPE_BU),
2               K_NCA_T = FINAL_STATE (DSC$K_CLASS_NCA, DSC$K_DTYPE_T),
2               K_VS_T = FINAL_STATE (DSC$K_CLASS_VS, DSC$K_DTYPE_T),
2               K_VS_VT = FINAL_STATE (DSC$K_CLASS_VS, DSC$K_DTYPE_VT),
2   !+
2   ! These are the left or right hand side of the conversion index.
2   !-
2               K_SMLINT = 1,
2               K_LRGINT = 2,
2               K_SMLFLT = 3,
2               K_LRGFLT = 4,
2               K_DEC = 5,
2               K_NBDS = 6,
2               K_TOT_CAT = 6;
2
2   !+
2   ! Define two structures.
2   ! START_STATE is just a vector of bytes, so we just use a macro to define it.
2   ! STATES is a structure that we put all the states in other than the first state,
2   ! and of course the final states and the states that never get used such as
2   ! the states that contain non-supported CLASSes will not be in this structure.
2   !-
2
2       STRUCTURE
2           STATES [STATE, TOKEN] =
2               [K_ACTUAL_CLASSES*K_MAX_DATA_TYPES]
4               (STATES + (K_MAX_DATA_TYPES*
5               BEGIN
5
5               CASE STATE FROM K_MIN_CLASS TO K_MAX_CLASS OF
5                   SET
5
5                   [K_STATE1_CLASS_S] :
5                       0;
5
5                   [K_STATE2_CLASS_D] :
5                       1;
5
5                   [K_STATE4_CLASS_A] :
5                       2;
5
5                   [K_STATE9_CLASS_SD] :
5                       3;
5
5                   [K_STATE10_CLASS_NCA] :
5                       4;
5
5                   [K_STATE11_CLASS_VS] :
5                       5;
5
5                   [INRANGE, OUTRANGE] :
6                       BEGIN
6                       LIB$STOP (LIB$_FATERRLIB);
6                       0
5                       END;
5                   TES
5
5               END
2               ) + TOKEN)<0, %BPUNIT, 1>;
2
2   !+
2   ! This is the start state entries.
2   ! For each CLASS in the standard there is an entry here.  They are :
2   !       Z       ,S      ,D      ,V      ,A
2   !      ,P       ,none   ,J      ,none   ,SD
2   !      ,NCA     ,VS     ,VSA    ,UBS    ,UBA.
2   !-
2
2       BIND
2           CLASS_TABLE = UPLIT BYTE
2         %( Start state.  All classes. )%
2         (K_UNSCLAROU,DSC$K_CLASS_S,DSC$K_CLASS_D,K_UNSCLAROU,DSC$K_CLASS_A
2         ,K_UNSCLAROU,K_UNSCLASTA,K_UNSCLAROU,K_UNSCLASTA,DSC$K_CLASS_SD
2         ,DSC$K_CLASS_NCA,DSC$K_CLASS_VS,K_UNSCLAROU,K_UNSCLAROU,K_UNSCLAROU) : START_STATE;
2
2   !+
2   ! This is the rest of the state table.  It is seperate because of space efficiency
2   ! Each state contains entries for each data type supported by the standard.
2   ! Note that for space efficiency The final states are not in the vector.
2   ! Also since each state represents a supported CLASS, if a CLASS is not
2   ! supported ( by the standard or routine ), then the state has no entry in
2   ! the vector.  The index table for the vector will index to the proper place
2   ! in the vector below.
2   ! This table shows graphically what descriptors are valid.
2   !                                  DSC$K__DTYPE__x
2   !                  BU WU LU B W L Q F D G H T NU NL NLO NR NRO NZ P VT
2   !DSC$K__CLASS__S   x  x  x  x x x x x x x x x x  x  x   x  x   x  x
2   !DSC$K__CLASS__D                              x
2   !DSC$K__CLASS__SD           x x x x x x x x x x  x  x   x  x   x  x
2   !DSC$K__CLASS__VS                             x                       x
2   !DSC$K__CLASS__A   x                          x
2   !DSC$K__CLASS__NCA x                          x
2   !
2   !
2   !Note that these data types are hard coded in ( zero based vector, and position
2   !of each data type is determined be the value of the symbol ) so if data type
2   !values are ever rearranged this table must be rearranged.
2   !-
2
2       BIND
2           DTYPE_TABLE = UPLIT BYTE
2         %( State zero. Class z. )%
2         %( State one.  Class s. )%
2         (K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_BU,DSC$K_DTYPE_WU,DSC$K_DTYPE_LU
2         ,K_UNSDTYROU,DSC$K_DTYPE_B,DSC$K_DTYPE_W,DSC$K_DTYPE_L,DSC$K_DTYPE_Q
2         ,DSC$K_DTYPE_F,DSC$K_DTYPE_D,K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_T
2         ,DSC$K_DTYPE_NU,DSC$K_DTYPE_NL,DSC$K_DTYPE_NLO,DSC$K_DTYPE_NR,DSC$K_DTYPE_NRO
2         ,DSC$K_DTYPE_NZ,DSC$K_DTYPE_P,K_UNSDTYROU,K_UNSDTYROU,K_UNSDESSTA
2         ,K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_G,DSC$K_DTYPE_H,K_UNSDTYROU
2         ,K_UNSDTYROU,K_UNSDTYROU,K_UNSDTYROU,K_UNSDTYROU,K_UNSDTYROU
2         ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
```

```
        %( State two.  Class d. )%
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,DSC$K_DTYPE_T
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        %( State three.  Class v. )%
        %( State four.  Class a. )%
        ,K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_BU,K_UNSDESROU,K_UNSDESROU
        ,K_UNSDTYROU,K_UNSDESROU,K_UNSDESROU,K_UNSDESROU,K_UNSDESROU
        ,K_UNSDESROU,K_UNSDESROU,K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_T
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDTYROU
        ,K_UNSDTYROU,K_UNSDTYROU,K_UNSDESROU,K_UNSDESROU,K_UNSDTYROU
        ,K_UNSDESSTA,K_UNSDTYROU,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA
        %( State five.  Class p. )%
        %( State six.  Class 'undefined' )%
        %( State seven.  Class j. )%
        %( State eight.  Class 'undefined' )%
        %( State nine.  Class sd. )%
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,DSC$K_DTYPE_B,DSC$K_DTYPE_W,DSC$K_DTYPE_L,DSC$K_DTYPE_Q
        ,DSC$K_DTYPE_F,DSC$K_DTYPE_D,K_UNSDESSTA,K_UNSDESSTA,DSC$K_DTYPE_T
        ,DSC$K_DTYPE_NU,DSC$K_DTYPE_NL,DSC$K_DTYPE_NLO,DSC$K_DTYPE_NR,DSC$K_DTYPE_NRO
        ,DSC$K_DTYPE_NZ,DSC$K_DTYPE_P,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDTYROU,DSC$K_DTYPE_G,DSC$K_DTYPE_H,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        %( State ten.  Class nca. )%
        ,K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_BU,K_UNSDESROU,K_UNSDESROU
        ,K_UNSDTYROU,K_UNSDESROU,K_UNSDESROU,K_UNSDESROU,K_UNSDESROU
        ,K_UNSDESROU,K_UNSDESROU,K_UNSDTYROU,K_UNSDTYROU,DSC$K_DTYPE_T
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDTYROU
        ,K_UNSDTYROU,K_UNSDTYROU,K_UNSDESROU,K_UNSDESROU,K_UNSDTYROU
        ,K_UNSDESSTA,K_UNSDTYROU,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA
        %( State eleven.  Class vs. )%
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,DSC$K_DTYPE_T
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA,K_UNSDESSTA
        ,K_UNSDESSTA,K_UNSDESSTA,DSC$K_DTYPE_VT
        %( State twelve.  Class vsa. )%
        %( State thirteen.  Class ubs. )%
        %( State fourteen.  Class uba. )%
        %( Add more states below )%
            ) : STATES;

    LOCAL
```

```
2               STATUS,                                 ! Status of this routine
2               STATE,                                  ! State
2               CLASS,                                  ! Current CLASS being looked at
2               DTYPE,                                  ! Current DTYPE being looked at
2               TOKEN,                                  ! The value of each data type supported
2               LEFT_CVT : VOLATILE VECTOR [1],         ! Left side of conversion index.
2               RIGHT_CVT : VOLATILE VECTOR [1],        ! Right side of conversion index.
2               LEFT_OR_RIGHT_CVT : REF VECTOR,         ! Left or right side of conversion index.
2               SRC_OR_DST_INFO : REF BLOCK [, BYTE],   ! Source or destination info.
2               SRC_OR_DST : REF BLOCK [, BYTE],        ! Source or destination.
2               TEMP_BUF : BLOCK [K_INTMED_DATA_LENGTH, BYTE];  ! Temporary buffer for reshuffling things.
2
2           MAP
2               SOURCE : REF BLOCK [, BYTE],
2               DESTINATION : REF BLOCK [, BYTE],
2               SRC_INFO : REF BLOCK [, BYTE] FIELD (SRC_INFO_FIELDS),
2               DST_INFO : REF BLOCK [, BYTE] FIELD (DST_INFO_FIELDS);
2
2       !+
2       ! Traverse through the state table twice.  Once for source, and once for
2       ! destination descriptor.
2       ! Each time come up with a final state that indicates which left hand side
2       ! (for the first traversing), or right hand side (for the second traversing) of
2       ! conversion we have got, e.g. SMLINT, or LRGFLT, etc.
2       ! The action codes also build SRC_INFO, and DST_INFO, and they do
2       ! the conversions to the intermediate values.
2       ! After we have the left hand side of conversion for source and the right hand
2       ! side of conversion for destination
2       ! descriptor, then stick them in a formula that maps these two into
2       ! one final answer that indicates which general CLASS, DTYPE is being
2       ! converted to which general CLASS, DTYPE, e.g. SMLINT LRGFLT, or DEC_SMLFLT, etc.
2       ! These final answers are the output parameter CVT_PATH that will end up as the
2       ! index to the CASE statement in LIB$CVT_DX_DX.
2       !-
2       !+
2       ! This loop is from 0 to 3, but we EXITLOOP at 2 because that is the second time
2       ! through the loop and the end of the road.
2       ! When the state table indicates an error, or we detect an error in an action routine,
2       ! we will just EXITLOOP with the value given by the state table, or of our own choice.
2       ! Note that we EXITLOOP when we detect errors in the action routines, e.g. if array
2       ! size is greater than a WU.
2       !-
3           BEGIN
4           STATUS = (INCRU TURN FROM 0 TO 3 DO
5               BEGIN
5       !+
5       ! Determine CLASS and DTYPE of this go around, also set up LEFT_OR_RIGHT_CVT,
5       ! and SRC_OR_DST, and SRC_OR_DST_INFO.
5       ! If this is the third time through this loop, we are finished.
5       !-
5
5               CASE .TURN FROM 0 TO 2 OF
5                   SET
5
5                   [0] :
6                       BEGIN
6                       CLASS = .SOURCE [DSC$B_CLASS];
```

```
6                       DTYPE = .SOURCE [DSC$B_DTYPE];
6                       SRC_OR_DST = .SOURCE;
6                       SRC_OR_DST_INFO = .SRC_INFO;
6                       LEFT_OR_RIGHT_CVT = LEFT_CVT;
5                       END;
5
5                   [1] :
6                       BEGIN
6                       CLASS = .DESTINATION [DSC$B_CLASS];
6                       DTYPE = .DESTINATION [DSC$B_DTYPE];
6                       SRC_OR_DST = .DESTINATION;
6                       SRC_OR_DST_INFO = .DST_INFO;
6                       LEFT_OR_RIGHT_CVT = RIGHT_CVT;
5                       END;
5
5                   [2] :
5                       EXITLOOP K_SUPPORTED;
5                   TES;
5
5   !+
5   ! Filter out the out-of-range CLASS and DTYPE.
5   !-
5
5               IF .CLASS GTRU K_MAX_CLASS_STA THEN EXITLOOP K_UNSCLASTA;
5
5               IF .DTYPE GTRU K_MAX_DTYPE_STA THEN EXITLOOP K_UNSDTYSTA;
5
5   !+
5   ! Crank up the finite state machine. start looking in the start state.
5   !-
5               STATE = .CLASS_TABLE [.CLASS];
5   !+
5   ! Action code for each state that results from the start state.
5   !-
5
5               CASE .STATE FROM K_MSTNEGERR TO K_LRGCLSSUP OF
5                   SET
5
5                   [K_INVNBDS TO K_UNSCLAROU] :
5                       EXITLOOP .STATE;                ! Exit the INCR with the error
5                                                       ! resulted from the start state.
5
5                   [K_SMLCLSSUP TO K_LRGCLSSUP] :
6                       BEGIN
6                       TOKEN = .DTYPE_TABLE [.STATE, .DTYPE];  ! This is a final state, but
6                                                       ! some constants need to be
6                                                       ! Applied to it yet.
6                                                       ! This is just a data type, or a negative number if error.
6
6                       IF .TOKEN LSS 0 THEN EXITLOOP .TOKEN;   ! Exit INCR with the error resulted
6
6                                                       ! in a final state.
6                       STATE = FINAL_STATE (.STATE, .TOKEN);   ! Find the final state.
5                       END;
5
5                   [INRANGE, OUTRANGE] :
5                       LIB$STOP (LIB$_FATERRLIB);
```

```
5                 TES;
5
5   !+
5   ! This CASE statement contains the action code for each final state other than
5   ! the erro states.
5   ! The caller of this routine has set up the pointer and length of SRC_INFO
5   ! to be the intermediate data area (INTMED_DATA), so in the CASE below we
5   ! will change pointer and length if needed (e.g. any NBDS), otherwise we never
5   ! touch it.
5   ! If .TURN is 0 then we are processing the left side of the conversion, when
5   ! it is 1 we are processing the right side of the conversion. Another words
5   ! if .TURN is 0 we are looking at the CLASS, DATA TYPE of source, and if
5   ! it is 1 we are looking at CLASS, DATA TYPE of destination.
5   ! These action codes determine which category (e.g. K_SMLINT or K_DEC as
5   ! described in LIB$CVT_DX_DX documentation) source or destination data type
5   ! falls into.  They also convert the source data type to an intermediate
5   ! data type.  For more detail refer to the functional description of
5   ! LIB$CVT_DX_DX.
5   !-
5
5           CASE .STATE FROM K_SMLFINSTA TO K_LRGFINSTA OF
5               SET
5
5               [K_S_BU, K_SD_BU] :
6                   BEGIN
6                   .LEFT_OR_RIGHT_CVT = K_SMLINT;
6
6                   IF .TURN EQL 0
6                   THEN
6                       .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 8, 0;,
6                           BYTE];
6
5                   END;
5
5               [K_S_WU, K_SD_WU] :
6                   BEGIN
6                   .LEFT_OR_RIGHT_CVT = K_SMLINT;
6
6                   IF .TURN EQL 0
6                   THEN
6                       .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 16, 0;,
6                           BYTE];
6
5                   END;
5
5               [K_S_LU, K_SD_LU] :
6                   BEGIN
6                   .LEFT_OR_RIGHT_CVT = K_LRGINT;
6
6                   IF .TURN EQL 0
6                   THEN
6                       .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 32, 0;,
6                           BYTE];
6
5                   END;
5
5               [K_S_B, K_SD_B] :
```

```
6                           BEGIN
6                           .LEFT_OR_RIGHT_CVT = K_SMLINT;
6
6                           IF .STATE EQL K_SD_B THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6
6                           IF .TURN EQL 0
6                           THEN
6                               .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 8, 1;,
6                                   BYTE];
6
6                           END;
5
5                       [K_S_W, K_SD_W] :
6                           BEGIN
6                           .LEFT_OR_RIGHT_CVT = K_SMLINT;
6
6                           IF .STATE EQL K_SD_W THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6
6                           IF .TURN EQL 0
6                           THEN
6                               .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 16, 1;,
6                                   BYTE];
6
5                           END;
5
5                       [K_S_L, K_SD_L] :
6                           BEGIN
6                           .LEFT_OR_RIGHT_CVT = K_SMLINT;
6
6                           IF .STATE EQL K_SD_L THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6
6                           IF .TURN EQL 0
6                           THEN
6                               .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 32, 1;,
6                                   BYTE];
6
5                           END;
5
5                       [K_S_Q, K_SD_Q] :
6                           BEGIN
6                           .LEFT_OR_RIGHT_CVT = K_LRGINT;
6
6                           IF .STATE EQL K_SD_Q THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6
6                           IF .TURN EQL 0
6                           THEN
7                               BEGIN
7                               .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 32, 0;, BYTE];
7                               (.SRC_INFO [S_POINTER] + 4) = .BLOCK [.SOURCE [DSC$A_POINTER] + 4, 0, 0, 32, 0;, BYTE];
7
7                               IF .BLOCK [.SRC_INFO [S_POINTER], 4, 31, 1, 0;, BYTE]
7                               THEN
8                                   BEGIN
8                                   .SRC_INFO [S_POINTER] = ..SRC_INFO [S_POINTER] XOR %X'FFFFFFFF';
8                                   .SRC_INFO [S_POINTER] + 4 = .(.SRC_INFO [S_POINTER] + 4) XOR %X'FFFFFFFF';
8
8                                   IF ..SRC_INFO [S_POINTER] EQLU K_LRGST_LU
```

```
                    THEN
                        BEGIN
                        .SRC_INFO [S_POINTER] = 0;
                        .SRC_INFO [S_POINTER] + 4 = .(.SRC_INFO [S_POINTER] + 4) + 1;
                        END
                    ELSE
                        .SRC_INFO [S_POINTER] = ..SRC_INFO [S_POINTER] + 1;

                    SRC_INFO [S_SIGN] = 1;
                    END;

                END;

            END;

        [K_S_F, K_SD_F] :
            BEGIN
            .LEFT_OR_RIGHT_CVT = K_SMLFLT;

            IF .STATE EQL K_SD_F THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

            IF .TURN EQL 0
            THEN
                .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 32, 0;,
                    BYTE];

            END;

        [K_S_D, K_SD_D] :
            BEGIN
            .LEFT_OR_RIGHT_CVT = K_SMLFLT;

            IF .STATE EQL K_SD_D THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

            IF .TURN EQL 0
            THEN
                BEGIN
                .SRC_INFO [S_POINTER] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 32, 0;, BYTE];
                (.SRC_INFO [S_POINTER] + 4) = .BLOCK [.SOURCE [DSC$A_POINTER] + 4, 0, 0, 32, 0;, BYTE];
                END;

            END;

        [K_S_G, K_SD_G] :
            BEGIN
            .LEFT_OR_RIGHT_CVT = K_LRGFLT;

            IF .STATE EQL K_SD_G THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

            IF .TURN EQL 0 THEN CVTGH (.SOURCE [DSC$A_POINTER], .SRC_INFO [S_POINTER]);

            END;

        [K_S_H, K_SD_H] :
            BEGIN
            .LEFT_OR_RIGHT_CVT = K_LRGFLT;
        IF .STATE EQL K_SD_H THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

        IF .TURN EQL 0 THEN CH$MOVE (16, .SOURCE [DSC$A_POINTER], .SRC_INFO [S_POINTER]);

        END;

    [K_S_T, K_SD_T] :
        BEGIN
        .LEFT_OR_RIGHT_CVT = K_NBDS;
        SRC_OR_DST_INFO [M_LEN] = .SRC_OR_DST [DSC$W_LENGTH];

        IF .STATE EQL K_SD_T THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

        IF .TURN EQL 0
        THEN
            BEGIN
            SRC_INFO [S_POINTER] = .SOURCE [DSC$A_POINTER];
            END;

        END;

    [K_S_NU, K_SD_NU] :
        BEGIN
        .LEFT_OR_RIGHT_CVT = K_DEC;

        IF .STATE EQL K_SD_NU THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

        IF .TURN EQL 0
        THEN
            BEGIN
            SRC_INFO [S_LEN] = 31;
            CVTTP (SOURCE [DSC$W_LENGTH], .SOURCE [DSC$A_POINTER], LIB$AB_CVTTP_U,
                SRC_INFO [S_LEN], .SRC_INFO [S_POINTER]);
            END;

        END;

    [K_S_NL, K_SD_NL] :
        BEGIN
        .LEFT_OR_RIGHT_CVT = K_DEC;

        IF .STATE EQL K_SD_NL THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];

        IF .TURN EQL 0
        THEN
            BEGIN
            SRC_INFO [S_LEN] = 31;
            CVTSP (%REF (

                    IF .SOURCE [DSC$W_LENGTH] EQL 0 THEN 0 ELSE .SOURCE [DSC$W_LENGTH] - 1),
                .SOURCE [DSC$A_POINTER], SRC_INFO [S_LEN], .SRC_INFO [S_POINTER]);

            END;

        END;

    [K_S_NLO, K_SD_NLO] :
```

```
6           BEGIN
6           .LEFT_OR_RIGHT_CVT = K_DEC;
6
6           IF .STATE EQL K_SD_NLO THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6
6           IF .TURN EQL 0
6           THEN
7               BEGIN
7
7               BIND FIRST_BYTE = SOURCE [DSC$A_POINTER] : REF VECTOR [,BYTE];
7
7               SRC_INFO [S_LEN] = 31;
7               CH$TRANSLATE (LIB$AB_CVT_O_U, .SOURCE [DSC$W_LENGTH], .SOURCE [DSC$A_POINTER], 0,
7                   .SOURCE [DSC$W_LENGTH], TEMP_BUF);
7               CVTTP (SOURCE [DSC$W_LENGTH], TEMP_BUF, LIB$AB_CVTTP_U, SRC_INFO [S_LEN],
7                   .SRC_INFO [S_POINTER]);
7
7               IF (.FIRST_BYTE [0] GEQU %X'4A' AND .FIRST_BYTE [0] LEQU %X'52') OR
7                   .FIRST_BYTE [0] EQLU %X'7D'
7               THEN
7                   BLOCK [.SRC_INFO [S_POINTER] + .SRC_INFO [S_LEN]/2, 0, 0, 4, 0;, BYTE] =
7                       .BLOCK [LIB$AB_CVTTP_O + .FIRST_BYTE [0], 0, 0, 4, 0;, BYTE];
7
6               END;
6
5           END;
5
5       [K_S_NR, K_SD_NR] :
6           BEGIN
6           .LEFT_OR_RIGHT_CVT = K_DEC;
6
6           IF .STATE EQL K_SD_NR THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6
6           IF .TURN EQL 0
6           THEN
7               BEGIN
7
7               LOCAL
7                   SOU_LEN;
7
7               SOU_LEN =
8               BEGIN
8
8               IF .SOURCE [DSC$W_LENGTH] EQL 0 THEN 0 ELSE .SOURCE [DSC$W_LENGTH] - 1
8
7               END;
7               TEMP_BUF [0, 0, 8, 0] = .BLOCK [.SOURCE [DSC$A_POINTER] + .SOU_LEN, 0, 0, 8, 0;, BYTE];
7               CH$MOVE (.SOU_LEN, .SOURCE [DSC$A_POINTER], TEMP_BUF + 1);
7               SRC_INFO [S_LEN] = 31;
7               CVTSP (SOU_LEN, TEMP_BUF, SRC_INFO [S_LEN], .SRC_INFO [S_POINTER]);
6               END;
6
5           END;
5
5       [K_S_NRO, K_SD_NRO] :
6           BEGIN
6           .LEFT_OR_RIGHT_CVT = K_DEC;
```

```
; 1012          1104  6
; 1013          1105  6                     IF .STATE EQL K_SD_NRO THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
; 1014          1106  6
; 1015          1107  6                     IF .TURN EQL 0
; 1016          1108  6                     THEN
; 1017          1109  7                         BEGIN
; 1018          1110  7                         SRC_INFO [S_LEN] = 31;
; 1019          1111  7                         CVTTP (SOURCE [DSC$W_LENGTH], .SOURCE [DSC$A_POINTER], LIB$AB_CVTTP_O,
; 1020          1112  7                             SRC_INFO [S_LEN], .SRC_INFO [S_POINTER]);
; 1021          1113  6                         END;
; 1022          1114  6
; 1023          1115  5                     END;
; 1024          1116  5
; 1025          1117  5                 [K_S_NZ, K_SD_NZ] :
; 1026          1118  6                     BEGIN
; 1027          1119  6                     .LEFT_OR_RIGHT_CVT = K_DEC;
; 1028          1120  6
; 1029          1121  6                     IF .STATE EQL K_SD_NZ THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
; 1030          1122  6
; 1031          1123  6                     IF .TURN EQL 0
; 1032          1124  6                     THEN
; 1033          1125  7                         BEGIN
; 1034          1126  7                         SRC_INFO [S_LEN] = 31;
; 1035          1127  7                         CVTTP (SOURCE [DSC$W_LENGTH], .SOURCE [DSC$A_POINTER], LIB$AB_CVTTP_Z,
; 1036          1128  7                             SRC_INFO [S_LEN], .SRC_INFO [S_POINTER]);
; 1037          1129  6                         END;
; 1038          1130  6
; 1039          1131  5                     END;
; 1040          1132  5
; 1041          1133  5                 [K_S_P, K_SD_P] :
; 1042          1134  6                     BEGIN
; 1043          1135  6                     .LEFT_OR_RIGHT_CVT = K_DEC;
; 1044          1136  6
; 1045          1137  6                     IF .STATE EQL K_SD_P THEN SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
; 1046          1138  6
; 1047          1139  6                     IF .TURN EQL 0
; 1048          1140  6                     THEN
; 1049          1141  7                         BEGIN
; 1050          1142  7                         CVTPS (SOURCE [DSC$W_LENGTH], .SOURCE [DSC$A_POINTER], %REF (31), TEMP_BUF);
; 1051          1143  7                         CVTSP (%REF (31), TEMP_BUF, %REF (31), .SRC_INFO [S_POINTER]);
; 1052          1144  7                         SRC_INFO [S_LEN] = 31;
; 1053          1145  6                         END;
; 1054          1146  6
; 1055          1147  5                     END;
; 1056          1148  5
; 1057          1149  5                 [K_D_T] :
; 1058          1150  6                     BEGIN
; 1059          1151  6                     .LEFT_OR_RIGHT_CVT = K_NBDS;
; 1060          1152  6                     SRC_OR_DST_INFO [M_LEN] = .SRC_OR_DST [DSC$W_LENGTH];
; 1061          1153  6
; 1062          1154  6                     IF .TURN EQL 0
; 1063          1155  6                     THEN
; 1064          1156  7                         BEGIN
; 1065          1157  7                         SRC_INFO [S_POINTER] = .SOURCE [DSC$A_POINTER];
; 1066          1158  6                         END;
; 1067          1159  6
; 1068          1160  5                     END;
```

```
5
5                [K_A_BU, K_A_T, K_NCA_BU, K_NCA_T] :
6                    BEGIN
6                    .LEFT_OR_RIGHT_CVT = K_NBDS;
6
7                    IF (.SRC_OR_DST [DSC$L_ARSIZE] GTR K_LRGST_WU OR .SRC_OR_DST [DSC$B_DIMCT] NEQ 1 OR
7                        .SRC_OR_DST [DSC$W_LENGTH] NEQ 1)
6                    THEN
6                        EXITLOOP K_INVNBDS;
6
7                    IF (.STATE EQL K_NCA_BU OR .STATE EQL K_NCA_T)
6                    THEN
7                        BEGIN
7
7                        IF .SRC_OR_DST [DSC$L_S1] NEQ 1 THEN EXITLOOP K_INVNBDS;
7
6                        END;
6
6                    SRC_OR_DST_INFO [M_SCALE] = .SRC_OR_DST [DSC$B_SCALE];
6                    SRC_OR_DST_INFO [M_LEN] = .SRC_OR_DST [DSC$L_ARSIZE];
6
6                    IF .TURN EQL 0
6                    THEN
7                        BEGIN
7                        SRC_INFO [S_POINTER] = .SOURCE [DSC$A_POINTER];
6                        END;
6
5                    END;
5
5                [K_VS_T, K_VS_VT] :
6                    BEGIN
6                    .LEFT_OR_RIGHT_CVT = K_NBDS;
6
6                    IF .TURN EQL 0
6                    THEN
7                        BEGIN
7                        SRC_INFO [S_POINTER] = .SOURCE [DSC$A_POINTER] + 2;
7                        SRC_INFO [S_LEN] = .BLOCK [.SOURCE [DSC$A_POINTER], 0, 0, 16, 0;, BYTE];
7                        END
6                    ELSE
6                        DST_INFO [D_LEN] = .DESTINATION [DSC$W_LENGTH];
6
5                    END;
5
5                [INRANGE] :
5                    LIB$STOP (LIB$_FATERRLIB);
5                TES;
5
5            END
4        )                                                   ! End of INCRU, with a EXITLOOP value.
2       END;                                                 ! End of STATUS.
2   !+
2   ! Map the left and right of the conversion, (i.e. if the conversion is
2   ! K_SMLINT_LRGFLT, then LEFT_CVT is SMLINT and RIGHT_CVT is LRGFLT)
2   ! into a final conversion index and return with the status of this routine.
2   !-
2       .CVT_PATH = (.LEFT_CVT - 1)*K_TOT_CAT + .RIGHT_CVT;
```

```
; 1126          1218  2     RETURN .STATUS;
; 1127          1219  1     END;                                                 ! End of routine LIB$$FIND_CVT_PATH

                                                                        .TITLE  LIB$$FIND_CVT_PATH LIB$$FIND_CVT_PATH  for inte
;                                                                                                  rnal use of LIB$CVT
                                                                        .IDENT  \1-006\

                                                                        .PSECT  _LIB$CODE,NOWRT,  SHR,  PIC,2

FF  FF  FF  0B  0A  09  FB  FF  FB  FF  04  FF  02  01  FF  00000 P.AAA: .BYTE  -1, 1, 2, -1, 4, -1, -5, -1, -5, 9, 10, -    ;
                                                                                11, -1, -1, -1
0E  FE  FE  0B  0A  09  08  07  06  FE  04  03  02  FE  FE  0000F P.AAB: .BYTE  -2, -2, 2, 3, 4, -2, 6, 7, 8, 9, 10, 11, -    ;
FE  1C  1B  FE  FE  FC  FE  FE  15  14  13  12  11  10  0F  0001E               -2, -2, 14, 15, 16, 17, 18, 19, 20, 21, -    ;
FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FE  FE  FE  FE  FE  0002D               -2, -2, -4, -2, -2, 27, 28, -2, -2, -2, -    ;
FC  FC  FC  FC  FC  FC  FC  0E  FC  FC  FC  FC  FC  FC  FC  0003C               -2, -2, -2, -4, -4, -4, -4, -4, -4, -4, -    ;
FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  0004B               -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
FE  FE  FD  FD  FD  FD  FD  FD  FE  FD  FD  02  FE  FE  FC  0005A               14, -4, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
FD  FD  FE  FE  FE  FC  FC  FC  FC  FC  FC  FC  FC  FC  0E  00069               -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FE  FC  FE  00078               -4, -4, -4, -4, -2, -2, 2, -3, -3, -2, -     ;
14  13  12  11  10  0F  0E  FC  FC  0B  0A  09  08  07  06  00087               -3, -3, -3, -3, -3, -3, -2, -2, 14, -4, -    ;
FC  FC  FC  FC  FC  FC  FC  1C  1B  FE  FC  FC  FC  FC  15  00096               -4, -4, -4, -4, -4, -4, -4, -4, -2, -2, -    ;
FE  FD  FD  FD  FD  FD  FD  FE  FD  FD  02  FE  FE  FC  FC  000A5               -2, -3, -3, -2, -4, -2, -4, -4, -4, -4, -    ;
FD  FE  FE  FE  FC  FC  FC  FC  FC  FC  FC  FC  FC  0E  FE  000B4               -4, -4, -4, -4, -4, -4, -4, -4, 6, 7, 8, -   ;
FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FE  FC  FE  FD  000C3               9, 10, 11, -4, -4, 14, 15, 16, 17, 18, -     ;
FC  FC  FC  FC  FC  0E  FC  FC  FC  FC  FC  FC  FC  FC  FC  000D2               19, 20, 21, -4, -4, -4, -4, -2, 27, 28, -    ;
FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  FC  000E1               -4, -4, -4, -4, -4, -4, -4, -4, -4, -2, -    ;
                                                25  FC  FC  000F0               -2, 2, -3, -3, -2, -3, -3, -3, -3, -3, -     ;
                                                                                -3, -2, -2, 14, -4, -4, -4, -4, -4, -4, -    ;
                                                                                -4, -4, -4, -2, -2, -2, -3, -3, -2, -4, -    ;
                                                                                -2, -4, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
                                                                                -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
                                                                                -4, 14, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
                                                                                -4, -4, -4, -4, -4, -4, -4, -4, -4, -4, -    ;
                                                                                -4, -4, -4, -4, 37                           ;

                                                                  CLASS_TABLE=          P.AAA
                                                                  DTYPE_TABLE=          P.AAB
                                                                        .EXTRN  LIB$STOP, LIB$$CVT_CVTGH_R1
                                                                        .EXTRN  LIB$AB_CVTTP_U, LIB$AB_CVT_O_U
                                                                        .EXTRN  LIB$AB_CVTTP_O, LIB$AB_CVT_U_O
                                                                        .EXTRN  LIB$AB_CVTPT_U, LIB$AB_CVTPT_O
                                                                        .EXTRN  LIB$AB_CVTPT_Z, LIB$AB_CVTTP_Z
                                                                        .EXTRN  LIB$_FATERRLIB

                                                       0FFC 00000       .ENTRY  LIB$$FIND_CVT_PATH, Save R2,R3,R4,R5,R6,R7,-; 0194
                                                                                R8,R9,R10,R11                                ;
                                  5E               38   C2 00002        SUBL2   #56, SP                                      ;
                                              0C   AE   D4 00005        CLRL    TURN                                         ; 0748
                    02            00          0C   AE   CF 00008 1$:    CASEL   TURN, #0, #2                                 ; 0756
                  003C          0021             0006      0000D 2$:    .WORD   3$-2$,-                                      ;
                                                                                4$-2$,-                                      ;
                                                                                5$-2$                                        ;
                                  50          04   AC   D0 00013 3$:    MOVL    SOURCE, R0                                   ; 0761
                          04      AE          03   A0   9A 00017        MOVZBL  3(R0), CLASS                                 ;
                                  6E          02   A0   9A 0001C        MOVZBL  2(R0), DTYPE                                 ; 0762
                                  58               50   D0 00020        MOVL    R0, SRC_OR_DST                               ; 0763
```

```
                                                                                        B  1
LIB$$FIND_CVT_P LIB$$FIND_CVT_PATH  for internal use of LIB$CVT 16-Sep-1984 00:54:19    VAX-11 Bliss-32 V4.0-742                Page  23
1-006           Deterministic Finite Automata for LIB$CVT_DX_DX 14-Sep-1984 12:38:50    [LIBRTL.SRC]LIBFINCVT.B32;1                  (3)

                                5B          0C    AC  D0 00023          MOVL    SRC_INFO, SRC_OR_DST_INFO                      ; 0764
                         08     AE          34    AE  9E 00027          MOVAB   LEFT_CVT, LEFT_OR_RIGHT_CVT                    ; 0765
                                                  20  11 0002C          BRB     6$                                             ; 0756
                                50          08    AC  D0 0002E 4$:      MOVL    DESTINATION, R0                                ; 0770
                         04     AE          03    A0  9A 00032          MOVZBL  3(R0), CLASS                                   ;
                                6E          02    A0  9A 00037          MOVZBL  2(R0), DTYPE                                   ; 0771
                                58                50  D0 0003B          MOVL    R0, SRC_OR_DST                                 ; 0772
                                5B          10    AC  D0 0003E          MOVL    DST_INFO, SRC_OR_DST_INFO                      ; 0773
                         08     AE          30    AE  9E 00042          MOVAB   RIGHT_CVT, LEFT_OR_RIGHT_CVT                   ; 0774
                                                  05  11 00047          BRB     6$                                             ; 0756
                                50                01  D0 00049 5$:      MOVL    #1, STATUS                                     ; 0778
                                                  5F  11 0004C          BRB     12$                                            ;
                                0E          04    AE  D1 0004E 6$:      CMPL    CLASS, #14                                     ; 0785
                                                  05  1B 00052          BLEQU   7$                                             ;
                                50                05  CE 00054          MNEGL   #5, STATUS                                     ;
                                                  54  11 00057          BRB     12$                                            ;
                                25                6E  D1 00059 7$:      CMPL    DTYPE, #37                                     ; 0787
                                                  05  1B 0005C          BLEQU   8$                                             ;
                                50                06  CE 0005E          MNEGL   #6, STATUS                                     ;
                                                  4A  11 00061          BRB     12$                                            ;
                                50          04    AE  D0 00063 8$:      MOVL    CLASS, R0                                      ; 0792
                                56        FEA1  CF40  98 00067          CVTBL   CLASS_TABLE[R0], STATE                         ;
                         12  FFFFFFF9       8F    56  CF 0006D          CASEL   STATE, #-7, #18                                ; 0797
        0035          0035            0035      0035     00075 9$:      .WORD   11$-9$,-                                       ;
        0026          0035            0035      0035     0007D                  11$-9$,-                                       ;
        003A          003A            003A      003A     00085                  11$-9$,-                                       ;
        003A          003A            003A      003A     0008D                  11$-9$,-                                       ;
                      003A            003A      003A     00095                  11$-9$,-                                       ;
                                                                                11$-9$,-                                       ;
                                                                                11$-9$,-                                       ;
                                                                                10$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$,-                                       ;
                                                                                13$-9$                                         ;
                                     00000000G    8F  DD 0009B 10$:     PUSHL   #LIB$_FATERRLIB                                ; 0818
                         00000000G  00            01  FB 000A1          CALLS   #1, LIB$STOP                                   ;
                                                  65  11 000A8          BRB     25$                                            ;
                                50                56  D0 000AA 11$:     MOVL    STATE, STATUS                                  ; 0801
                                                  55  11 000AD 12$:     BRB     23$                                            ;
                        0A      01                56  CF 000AF 13$:     CASEL   STATE, #1, #10                                 ; 0806
        002C          0016            0027      0023     000B3 14$:     .WORD   16$-14$,-                                      ;
        0016          0016            0016      0016     000BB                  17$-14$,-                                      ;
                      003B            0036      0031     000C3                  15$-14$,-                                      ;
                                                                                18$-14$,-                                      ;
                                                                                15$-14$,-                                      ;
                                                                                15$-14$,-                                      ;
                                                                                15$-14$,-                                      ;
                                                                                15$-14$,-                                      ;
                                                                                19$-14$,-                                      ;
```

```
                                                                  C  1
LIB$$FIND_CVT_P LIB$$FIND_CVT_PATH  for internal use of LIB$CVT 16-Sep-1984 00:54:19    VAX-11 Bliss-32 V4.0-742              Page  24
1-006           Deterministic Finite Automata for LIB$CVT_DX_DX 14-Sep-1984 12:38:50    [LIBRTL.SRC]LIBFINCVT.B32;1                (3)

                                                                    20$-14$,-                              ;
                                                                    21$-14$                                ;
                                   00000000G  8F  DD 000C9 15$:     PUSHL   #LIB$_FATERRLIB                ;
                     00000000G  00            01  FB 000CF          CALLS   #1, LIB$STOP                   ;
                                              50  D4 000D6 16$:     CLRL    R0                             ;
                                              17  11 000D8          BRB     22$                            ;
                                50            01  D0 000DA 17$:     MOVL    #1, R0                         ;
                                              12  11 000DD          BRB     22$                            ;
                                50            02  D0 000DF 18$:     MOVL    #2, R0                         ;
                                              0D  11 000E2          BRB     22$                            ;
                                50            03  D0 000E4 19$:     MOVL    #3, R0                         ;
                                              08  11 000E7          BRB     22$                            ;
                                50            04  D0 000E9 20$:     MOVL    #4, R0                         ;
                                              03  11 000EC          BRB     22$                            ;
                                50            05  D0 000EE 21$:     MOVL    #5, R0                         ;
                                50            26  C4 000F1 22$:     MULL2   #38, R0                        ;
                                50     FE23 CF40  9E 000F4          MOVAB   DTYPE_TABLE[R0], R0            ;
                                57       00 BE40  98 000FA          CVTBL   @DTYPE[R0], TOKEN              ;
                                              06  18 000FF          BGEQ    24$                            ; 0811
                                50            57  D0 00101          MOVL    TOKEN, STATUS                  ;
                                            073C  31 00104 23$:     BRW     105$                           ;
                      50        56            26  C5 00107 24$:     MULL3   #38, STATE, R0                 ; 0814
                      56        50            57  C1 0010B          ADDL3   TOKEN, R0, STATE               ;
        0000019F      8F        28            56  CF 0010F 25$:     CASEL   STATE, #40, #415               ; 0839
   0710       0370      0358        0340         00117 26$:     .WORD   27$-26$,-                      ;
   03CF       03C0      039B        0376         0011F              28$-26$,-                      ;
   0710       0710      0444        041F         00127              29$-26$,-                      ;
   0545       050C      04DB        04C3         0012F              102$-26$,-                     ;
   0658       0627      05F6        05B1         00137              30$-26$,-                      ;
   0710       0710      0710        0710         0013F              32$-26$,-                      ;
   0710       049D      046F        0710         00147              35$-26$,-                      ;
   0710       0710      0710        0710         0014F              37$-26$,-                      ;
   0710       0710      0710        0710         00157              43$-26$,-                      ;
   0710       0710      0710        0710         0015F              46$-26$,-                      ;
   0710       0710      0710        0710         00167              102$-26$,-                     ;
   0710       0710      0710        0710         0016F              102$-26$,-                     ;
   0710       068A      0710        0710         00177              57$-26$,-                      ;
   0710       0710      0710        0710         0017F              59$-26$,-                      ;
   0710       0710      0710        0710         00187              63$-26$,-                      ;
   0710       0710      0710        0710         0018F              69$-26$,-                      ;
   0710       0710      0710        0710         00197              76$-26$,-                      ;
   0710       0710      0710        0710         0019F              81$-26$,-                      ;
   0710       0710      0710        0710         001A7              85$-26$,-                      ;
   0710       0710      0710        0710         001AF              89$-26$,-                      ;
   0710       0710      0710        0710         001B7              102$-26$,-                     ;
   0710       0710      0710        0710         001BF              102$-26$,-                     ;
   0710       0710      0710        0710         001C7              102$-26$,-                     ;
   0710       0710      0710        0710         001CF              102$-26$,-                     ;
   0710       0710      0710        0710         001D7              102$-26$,-                     ;
   0710       0710      0710        0710         001DF              50$-26$,-                      ;
   0710       0710      0710        0710         001E7              53$-26$,-                      ;
   0710       0710      0710        0710         001EF              102$-26$,-                     ;
   0710       0694      0710        0710         001F7              102$-26$,-                     ;
   0710       0710      0710        0710         001FF              102$-26$,-                     ;
   0710       0710      0710        0710         00207              102$-26$,-                     ;
   0710       0694      0710        0710         0020F              102$-26$,-                     ;
   0710       0710      0710        0710         00217              102$-26$,-                     ;
```

```
      0710        0710        0710        0710     0021F            102$-26$,-              :
      0710        0710        0710        0710     00227            102$-26$,-              :
      0710        0710        0710        0710     0022F            102$-26$,-              :
      0710        0710        0710        0710     00237            102$-26$,-              :
      0710        0710        0710        0710     0023F            102$-26$,-              :
      0710        0710        0710        0710     00247            102$-26$,-              :
      0710        0710        0710        0710     0024F            102$-26$,-              :
      0710        0710        0710        0710     00257            102$-26$,-              :
      0710        0710        0710        0710     0025F            102$-26$,-              :
      0710        0710        0710        0710     00267            102$-26$,-              :
      0710        0710        0710        0710     0026F            102$-26$,-              :
      0710        0710        0710        0710     00277            102$-26$,-              :
      0710        0710        0710        0710     0027F            102$-26$,-              :
      0710        0710        0710        0710     00287            102$-26$,-              :
      0710        0710        0710        0710     0028F            102$-26$,-              :
      0710        0710        0710        0710     00297            102$-26$,-              :
      0710        0710        0710        0710     0029F            102$-26$,-              :
      0710        0710        0710        0710     002A7            92$-26$,-               :
      0710        0710        0710        0710     002AF            102$-26$,-              :
      0710        0710        0710        0710     002B7            102$-26$,-              :
      0710        0710        0710        0710     002BF            102$-26$,-              :
      0710        0710        0710        0710     002C7            102$-26$,-              :
      0710        0710        0710        0710     002CF            102$-26$,-              :
      0710        0710        0710        0710     002D7            102$-26$,-              :
      0710        0710        0710        0710     002DF            102$-26$,-              :
      0710        0710        0710        0710     002E7            102$-26$,-              :
      0710        0710        0710        0710     002EF            102$-26$,-              :
      0710        0710        0710        0710     002F7            102$-26$,-              :
      0710        0710        0710        0710     002FF            102$-26$,-              :
      0710        0710        0710        0710     00307            102$-26$,-              :
      0710        0710        0710        0710     0030F            102$-26$,-              :
      0710        0710        0710        0710     00317            102$-26$,-              :
      0710        0710        0710        0710     0031F            102$-26$,-              :
      0710        0710        0710        0710     00327            102$-26$,-              :
      0710        0710        0710        0710     0032F            102$-26$,-              :
      0710        0710        0710        0710     00337            102$-26$,-              :
      0710        0710        0710        0710     0033F            102$-26$,-              :
      0710        0710        0710        0710     00347            102$-26$,-              :
      0710        0710        0710        0710     0034F            102$-26$,-              :
      0710        0710        0710        0710     00357            102$-26$,-              :
      0710        0710        0710        0710     0035F            102$-26$,-              :
      0710        0710        0710        0710     00367            102$-26$,-              :
      0710        0710        0710        0710     0036F            102$-26$,-              :
      0710        0370        0358        0340     00377            102$-26$,-              :
      03CF        03C0        039B        0376     0037F            102$-26$,-              :
      0710        0710        0444        041F     00387            102$-26$,-              :
      0545        050C        04DB        04C3     0038F            102$-26$,-              :
      0658        0627        05F6        05B1     00397            102$-26$,-              :
      0710        0710        0710        0710     0039F            102$-26$,-              :
      0710        049D        046F        0710     003A7            102$-26$,-              :
      0710        0710        0710        0710     003AF            102$-26$,-              :
      0710        0710        0710        0710     003B7            102$-26$,-              :
      0710        0694        0710        0710     003BF            102$-26$,-              :
      0710        0710        0710        0710     003C7            102$-26$,-              :
      0710        0710        0710        0710     003CF            102$-26$,-              :
      0710        0694        0710        0710     003D7            102$-26$,-              :
      0710        0710        0710        0710     003DF            102$-26$,-              :
        0710          0710          0710           0710     003E7                 102$-26$,-
        0710          0710          0710           0710     003EF                 102$-26$,-
        0710          0710          0710           0710     003F7                 102$-26$,-
        0710          0710          0710           0710     003FF                 102$-26$,-
        0710          0710          0710           0710     00407                 102$-26$,-
        0710          0710          0710           0710     0040F                 102$-26$,-
        0710          0710          0710           0710     00417                 102$-26$,-
        0710          0710          0710           0710     0041F                 102$-26$,-
        0710          0710          0710           06E7     00427                 102$-26$,-
        0710          0710          0710           0710     0042F                 102$-26$,-
        0710          0710          0710           0710     00437                 102$-26$,-
        0710          0710          0710           0710     0043F                 102$-26$,-
        0710          0710          0710           0710     00447                 102$-26$,-
        06E7          0710          0710           0710     0044F                 102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  93$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  93$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
                                                                                  102$-26$,-
```

```
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
                                                                        102$-26$,-                                     ;
```

```
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
```

```
102$-26$,-
1 2$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
102$-26$,-
27$-26$,-
28$-26$,-
29$-26$,-
102$-26$,-
30$-26$,-
32$-26$,-
35$-26$,-
37$-26$,-
43$-26$,-
46$-26$,-
102$-26$,-
102$-26$,-
57$-26$,-
59$-26$,-
```

```
                                                                  63$-26$,-                                           :
                                                                  69$-26$,-                                           :
                                                                  76$-26$,-                                           :
                                                                  81$-26$,-                                           :
                                                                  85$-26$,-                                           :
                                                                  89$-26$,-                                           :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  50$-26$,-                                           :
                                                                  53$-26$,-                                           :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  93$-26$,-                                           :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  93$-26$,-                                           :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                  102$-26$,-                                          :
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    99$-26$,-                                      ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    102$-26$,-                                     ;
                                                                    99$-26$                                        ;
                 08   BE          01  D0 00457 27$:     MOVL    #1, @LEFT_OR_RIGHT_CVT                             ; 0844
                           0C     AE  D5 0045B          TSTL    TURN                                               ; 0846
                                  75  12 0045E          BNEQ    34$                                                ;
                      51   0C     AC  D0 00460          MOVL    SRC_INFO, R1                                       ; 0848
                      50   04     AC  D0 00464          MOVL    SOURCE, R0                                         ;
                 01   B1   04     B0  9A 00468          MOVZBL  @4(R0), @1(R1)                                     ;
                                  66  11 0046D          BRB     34$                                                ; 0839
                 08   BE          01  D0 0046F 28$:     MOVL    #1, @LEFT_OR_RIGHT_CVT                             ; 0855
                           0C     AE  D5 00473          TSTL    TURN                                               ; 0857
                                  5D  12 00476          BNEQ    34$                                                ;
                      51   0C     AC  D0 00478          MOVL    SRC_INFO, R1                                       ; 0859
                      50   04     AC  D0 0047C          MOVL    SOURCE, R0                                         ;
                 01   B1   04     B0  3C 00480          MOVZWL  @4(R0), @1(R1)                                     ;
                                  4E  11 00485          BRB     34$                                                ; 0839
                 08   BE          02  D0 00487 29$:     MOVL    #2, @LEFT_OR_RIGHT_CVT                             ; 0866
                                  57  11 0048B          BRB     36$                                                ; 0868
```

```
             08    BE          01 D0 0048D 30$:     MOVL    #1, @LEFT_OR_RIGHT_CVT                        ; 0877
       0000015C    8F          56 D1 00491          CMPL    STATE, #348                                   ; 0879
                               04 12 00498          BNEQ    31$                                           ;
                   6B    08    A8 90 0049A          MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)              ;
                         0C    AE D5 0049E 31$:     TSTL    TURN                                          ; 0881
                               57 12 004A1          BNEQ    39$                                           ;
                   51    0C    AC D0 004A3          MOVL    SRC_INFO, R1                                  ; 0883
                   50    04    AC D0 004A7          MOVL    SOURCE, R0                                    ;
             01    B1    04    B0 98 004AB          CVTBL   @4(R0), @1(R1)                                ;
                               23 11 004B0          BRB     34$                                           ; 0839
             08    BE          01 D0 004B2 32$:     MOVL    #1, @LEFT_OR_RIGHT_CVT                        ; 0890
       0000015D    8F          56 D1 004B6          CMPL    STATE, #349                                   ; 0892
                               04 12 004BD          BNEQ    33$                                           ;
                   6B    08    A8 90 004BF          MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)              ;
                         0C    AE D5 004C3 33$:     TSTL    TURN                                          ; 0894
                               6C 12 004C6          BNEQ    42$                                           ;
                   51    0C    AC D0 004C8          MOVL    SRC_INFO, R1                                  ; 0896
                   50    04    AC D0 004CC          MOVL    SOURCE, R0                                    ;
             01    B1    04    B0 32 004D0          CVTWL   @4(R0), @1(R1)                                ;
                               5D 11 004D5 34$:     BRB     42$                                           ; 0839
             08    BE          01 D0 004D7 35$:     MOVL    #1, @LEFT_OR_RIGHT_CVT                        ; 0903
       0000015E    8F          56 D1 004DB          CMPL    STATE, #350                                   ; 0905
                               5F 13 004E2          BEQL    44$                                           ;
                               61 11 004E4 36$:     BRB     45$                                           ; 0907
             08    BE          02 D0 004E6 37$:     MOVL    #2, @LEFT_OR_RIGHT_CVT                        ; 0916
       0000015F    8F          56 D1 004EA          CMPL    STATE, #351                                   ; 0918
                               04 12 004F1          BNEQ    38$                                           ;
                   6B    08    A8 90 004F3          MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)              ;
                         0C    AE D5 004F7 38$:     TSTL    TURN                                          ; 0920
                               73 12 004FA 39$:     BNEQ    48$                                           ;
                   53    0C    AC D0 004FC          MOVL    SRC_INFO, R3                                  ; 0923
                   50    01    A3 D0 00500          MOVL    1(R3), R0                                     ;
                   51    04    AC D0 00504          MOVL    SOURCE, R1                                    ;
                   52    04    A1 D0 00508          MOVL    4(R1), R2                                     ;
                   60          62 D0 0050C          MOVL    (R2), (R0)                                    ;
                   51    04    A0 9E 0050F          MOVAB   4(R0), R1                                     ; 0924
                   61    04    A2 D0 00513          MOVL    4(R2), (R1)                                   ;
                               6B 18 00517          BGEQ    49$                                           ; 0926
                   60          60 D2 00519          MCOML   (R0), (R0)                                    ; 0929
                   61          61 D2 0051C          MCOML   (R1), (R1)                                    ; 0930
       FFFFFFFF    8F          60 D1 0051F          CMPL    (R0), #-1                                     ; 0932
                               06 12 00526          BNEQ    40$                                           ;
                               60 D4 00528          CLRL    (R0)                                          ; 0935
                               61 D6 0052A          INCL    (R1)                                          ; 0936
                               02 11 0052C          BRB     41$                                           ; 0932
                               60 D6 0052E 40$:     INCL    (R0)                                          ; 0939
             07    A3          01 88 00530 41$:     BISB2   #1, 7(R3)                                     ; 0941
                               7C 11 00534 42$:     BRB     52$                                           ; 0839
             08    BE          03 D0 00536 43$:     MOVL    #3, @LEFT_OR_RIGHT_CVT                        ; 0950
       00000160    8F          56 D1 0053A          CMPL    STATE, #352                                   ; 0952
                               04 12 00541          BNEQ    45$                                           ;
                   6B    08    A8 90 00543 44$:     MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)              ;
                         0C    AE D5 00547 45$:     TSTL    TURN                                          ; 0954
                               7C 12 0054A          BNEQ    55$                                           ;
                   51    0C    AC D0 0054C          MOVL    SRC_INFO, R1                                  ; 0956
                   50    04    AC D0 00550          MOVL    SOURCE, R0                                    ;
             01    B1    04    B0 D0 00554          MOVL    @4(R0), @1(R1)                                ;
```

```
                                     7D  11 00559       BRB     56$                                             : 0839
                       08    BE      03  D0 0055B 46$:  MOVL    #3, @LEFT_OR_RIGHT_CVT                          : 0963
                 00000161    8F      56  D1 0055F       CMPL    STATE, #353                                     : 0965
                                     04  12 00566       BNEQ    47$                                             :
                             6B  08  A8  90 00568       MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                :
                                 0C  AE  D5 0056C 47$:  TSTL    TURN                                            : 0967
                                     67  12 0056F 48$:  BNEQ    56$                                             :
                             50  0C  AC  D0 00571       MOVL    SRC_INFO, R0                                    : 0970
                             51  01  A0  D0 00575       MOVL    1(R0), R1                                       :
                             50  04  AC  D0 00579       MOVL    SOURCE, R0                                      :
                             50  04  A0  D0 0057D       MOVL    4(R0), R0                                       :
                             61      60  7D 00581       MOVQ    (R0), (R1)                                      :
                                     52  11 00584 49$:  BRB     56$                                             : 0839
                       08    BE      04  D0 00586 50$:  MOVL    #4, @LEFT_OR_RIGHT_CVT                          : 0978
                 00000171    8F      56  D1 0058A       CMPL    STATE, #369                                     : 0980
                                     04  12 00591       BNEQ    51$                                             :
                             6B  08  A8  90 00593       MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                :
                                 0C  AE  D5 00597 51$:  TSTL    TURN                                            : 0982
                                     6A  12 0059A       BNEQ    61$                                             :
                             53  0C  AC  D0 0059C       MOVL    SRC_INFO, R3                                    :
                             52  04  AC  D0 005A0       MOVL    SOURCE, R2                                      :
                             51  01  A3  D0 005A4       MOVL    1(R3), R1                                       :
                             50  04  A2  D0 005A8       MOVL    4(R2), R0                                       :
                            00000000G  00  16 005AC     JSB     LIB$$CVT_CVTGH_R1                               :
                                     6D  11 005B2 52$:  BRB     62$                                             : 0839
                       08    BE      04  D0 005B4 53$:  MOVL    #4, @LEFT_OR_RIGHT_CVT                          : 0988
                 00000172    8F      56  D1 005B8       CMPL    STATE, #370                                     : 0990
                                     04  12 005BF       BNEQ    54$                                             :
                             6B  08  A8  90 005C1       MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                :
                                 0C  AE  D5 005C5 54$:  TSTL    TURN                                            : 0992
                                     6D  12 005C8 55$:  BNEQ    65$                                             :
                             51  04  AC  D0 005CA       MOVL    SOURCE, R1                                      :
                             50  0C  AC  D0 005CE       MOVL    SRC_INFO, R0                                    :
          01    B0     04    B1      10  28 005D2       MOVC3   #16, @4(R1), @1(R0)                             :
                                     47  11 005D8 56$:  BRB     62$                                             : 0839
                       08    BE      06  D0 005DA 57$:  MOVL    #6, @LEFT_OR_RIGHT_CVT                          : 0998
                       05    AB      68  B0 005DE       MOVW    (SRC_OR_DST), 5(SRC_OR_DST_INFO)                : 0999
                 00000164    8F      56  D1 005E2       CMPL    STATE, #356                                     : 1001
                                     04  12 005E9       BNEQ    58$                                             :
                             6B  08  A8  90 005EB       MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                :
                                   01F8  31 005EF 58$:  BRW     97$                                             : 1003
                       08    BE      05  D0 005F2 59$:  MOVL    #5, @LEFT_OR_RIGHT_CVT                          : 1013
                 00000165    8F      56  D1 005F6       CMPL    STATE, #357                                     : 1015
                                     04  12 005FD       BNEQ    60$                                             :
                             6B  08  A8  90 005FF       MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                :
                                 0C  AE  D5 00603 60$:  TSTL    TURN                                            : 1017
                                     68  12 00606 61$:  BNEQ    71$                                             :
                             50  0C  AC  D0 00608       MOVL    SRC_INFO, R0                                    : 1020
                       05    A0      1F  B0 0060C       MOVW    #31, 5(R0)                                      :
                             51  04  AC  D0 00610       MOVL    SOURCE, R1                                      : 1021
  05    A0 00000000G  00   04    B1      61  26 00614   CVTTP   (R1), @4(R1), LIB$AB_CVTTP_U, 5(R0), @1(R0)     : 1022
                                 01  B0     0061F                                                               :
                                     37  11 00621 62$:  BRB     68$                                             : 0839
                       08    BE      05  D0 00623 63$:  MOVL    #5, @LEFT_OR_RIGHT_CVT                          : 1029
                 00000166    8F      56  D1 00627       CMPL    STATE, #358                                     : 1031
                                     04  12 0062E       BNEQ    64$                                             :
                             6B  08  A8  90 00630       MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                :
```

```
                                                                     M 1
LIB$$FIND_CVT_P LIB$$FIND_CVT_PATH  for internal use of LIB$CVT 16-Sep-1984 00:54:19     VAX-11 Bliss-32 V4.0-742         Page  34
1-006            Deterministic Finite Automata for LIB$CVT_DX_DX 14-Sep-1984 12:38:50     [LIBRTL.SRC]LIBFINCVT.B32;1         (3)

                                                         0C    AE D5 00634 64$:   TSTL    TURN                                          ; 1033
                                                               76 12 00637 65$:   BNEQ    73$                                           ;
                                               50        0C    AC D0 00639        MOVL    SRC_INFO, R0                                  ; 1036
                                         05    A0              1F B0 0063D        MOVW    #31, 5(R0)                                    ;
                                               51        04    AC D0 00641        MOVL    SOURCE, R1                                    ; 1039
                                                               61 B5 00645        TSTW    (R1)                                          ;
                                                               04 12 00647        BNEQ    66$                                           ;
                                                               52 D4 00649        CLRL    R2                                            ;
                                                               05 11 0064B        BRB     67$                                           ;
                                               52              61 3C 0064D 66$:   MOVZWL  (R1), R2                                      ;
                                                               52 D7 00650        DECL    R2                                            ;
        01    B0         05    A0        04    B1              52 09 00652 67$:   CVTSP   R2, @4(R1), 5(R0), @1(R0)                     ; 1040
                                                               6A 11 0065A 68$:   BRB     75$                                           ; 0839
                                         08    BE              05 D0 0065C 69$:   MOVL    #5, @LEFT_OR_RIGHT_CVT                        ; 1048
                                   00000167    8F              56 D1 00660        CMPL    STATE, #359                                   ; 1050
                                                               04 12 00667        BNEQ    70$                                           ;
                                               6B        08    A8 90 00669        MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)              ;
                                                         0C    AE D5 0066D 70$:   TSTL    TURN                                          ; 1052
                                                               6A 12 00670 71$:   BNEQ    78$                                           ;
                                               5A        04    AC D0 00672        MOVL    SOURCE, R10                                   ; 1056
                                               59        0C    AC D0 00676        MOVL    SRC_INFO, R9                                  ; 1058
                                         05    A9              1F B0 0067A        MOVW    #31, 5(R9)                                    ;
 00000000G    00               00        04    BA        04    BC 2E 0067E        MOVTC   @SOURCE, @4(R10), #0, LIB$AB_CVT_O_U, -       ; 1059
                                         10    AE        04    BC    00689                @SOURCE, TEMP_BUF                             ;
        05    A9  00000000G    00        10    AE        04    BC 26 0068D        CVTTP   @SOURCE, TEMP_BUF, LIB$AB_CVTTP_U, 5(R9), -   ; 1062
                                                         01    B9    00699                @1(R9)                                        ;
                                               51        04    BA 9A 0069B        MOVZBL  @4(R10), R1                                   ; 1064
                                         4A    8F              51 91 0069F        CMPB    R1, #74                                       ;
                                                               06 1F 006A3        BLSSU   72$                                           ;
                                         52    8F              51 91 006A5        CMPB    R1, #82                                       ;
                                                               06 1B 006A9        BLEQU   74$                                           ;
                                         7D    8F              51 91 006AB 72$:   CMPB    R1, #125                                      ; 1065
                                                               70 12 006AF 73$:   BNEQ    83$                                           ;
                                               50        05    A9 3C 006B1 74$:   MOVZWL  5(R9), R0                                     ; 1067
                                               50              02 C6 006B5        DIVL2   #2, R0                                        ;
                                                   00000000G00041 9F 006B8        PUSHAB  LIB$AB_CVTTP_O[R1]                            ; 1068
         01 B940               04              00              9E F0 006BF        INSV    @(SP)+, #0, #4, @1(R9)[R0]                    ;
                                                               74 11 006C6 75$:   BRB     84$                                           ; 0839
                                         08    BE              05 D0 006C8 76$:   MOVL    #5, @LEFT_OR_RIGHT_CVT                        ; 1076
                                   00000168    8F              56 D1 006CC        CMPL    STATE, #360                                   ; 1078
                                                               04 12 006D3        BNEQ    77$                                           ;
                                               6B        08    A8 90 006D5        MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)              ;
                                                         0C    AE D5 006D9 77$:   TSTL    TURN                                          ; 1080
                                                               74 12 006DC 78$:   BNEQ    87$                                           ;
                                               50        04    AC D0 006DE        MOVL    SOURCE, R0                                    ; 1090
                                                               60 B5 006E2        TSTW    (R0)                                          ;
                                                               04 12 006E4        BNEQ    79$                                           ;
                                                               5A D4 006E6        CLRL    SOU_LEN                                       ;
                                                               05 11 006E8        BRB     80$                                           ;
                                               5A              60 3C 006EA 79$:   MOVZWL  (R0), SOU_LEN                                 ;
                                                               5A D7 006ED        DECL    SOU_LEN                                       ;
                                         10    AE         04 B04A 90 006EF 80$:   MOVB    @4(R0)[SOU_LEN], TEMP_BUF                     ; 1093
                         11    AE        04    B0              5A 28 006F5        MOVC3   SOU_LEN, @4(R0), TEMP_BUF+1                   ; 1094
                                               50        0C    AC D0 006FB        MOVL    SRC_INFO, R0                                  ; 1095
                                         05    A0              1F B0 006FF        MOVW    #31, 5(R0)                                    ;
        01    B0         05    A0        10    AE              5A 09 00703        CVTSP   SOU_LEN, TEMP_BUF, 5(R0), @1(R0)              ; 1096
                                                               60 11 0070B        BRB     88$                                           ; 0839
```

```
                                08   BE            05  D0 0070D 81$:     MOVL    #5, @LEFT_OR_RIGHT_CVT                          ; 1103
                          00000169   8F            56  D1 00711          CMPL    STATE, #361                                     ; 1105
                                                   04  12 00718          BNEQ    82$                                             ;
                                6B            08   A8  90 0071A          MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                ;
                                              0C   AE  D5 0071E 82$:     TSTL    TURN                                            ; 1107
                                                   7C  12 00721 83$:     BNEQ    91$                                             ;
                                50            0C   AC  D0 00723          MOVL    SRC_INFO, R0                                    ; 1110
                          05    A0                 1F  B0 00727          MOVW    #31, 5(R0)                                      ;
                                51            04   AC  D0 0072B          MOVL    SOURCE, R1                                      ; 1111
      05   A0 00000000G   00    04   B1            61  26 0072F          CVTTP   (R1), @4(R1), LIB$AB_CVTTP_O, 5(R0), @1(R0)     ; 1112
                                              01   B0     0073A                                                                  ;
                                                   61  11 0073C 84$:     BRB     91$                                             ; 0839
                                08   BE            05  D0 0073E 85$:     MOVL    #5, @LEFT_OR_RIGHT_CVT                          ; 1119
                          0000016A   8F            56  D1 00742          CMPL    STATE, #362                                     ; 1121
                                                   04  12 00749          BNEQ    86$                                             ;
                                6B            08   A8  90 0074B          MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                ;
                                              0C   AE  D5 0074F 86$:     TSTL    TURN                                            ; 1123
                                                   4B  12 00752 87$:     BNEQ    91$                                             ;
                                50            0C   AC  D0 00754          MOVL    SRC_INFO, R0                                    ; 1126
                          05    A0                 1F  B0 00758          MOVW    #31, 5(R0)                                      ;
                                51            04   AC  D0 0075C          MOVL    SOURCE, R1                                      ; 1127
      05   A0 00000000G   00    04   B1            61  26 00760          CVTTP   (R1), @4(R1), LIB$AB_CVTTP_Z, 5(R0), @1(R0)     ; 1128
                                              01   B0     0076B                                                                  ;
                                                   30  11 0076D 88$:     BRB     91$                                             ; 0839
                                08   BE            05  D0 0076F 89$:     MOVL    #5, @LEFT_OR_RIGHT_CVT                          ; 1135
                          0000016B   8F            56  D1 00773          CMPL    STATE, #363                                     ; 1137
                                                   04  12 0077A          BNEQ    90$                                             ;
                                6B            08   A8  90 0077C          MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                ;
                                              0C   AE  D5 00780 90$:     TSTL    TURN                                            ; 1139
                                                   77  12 00783          BNEQ    98$                                             ;
                                50            04   AC  D0 00785          MOVL    SOURCE, R0                                      ; 1142
      10   AE             1F    04   B0            60  08 00789          CVTPS   (R0), @4(R0), #31, TEMP_BUF                     ;
                                54            0C   AC  D0 00790          MOVL    SRC_INFO, R4                                    ; 1143
      01   B4             1F    10   AE            1F  09 00794          CVTSP   #31, TEMP_BUF, #31, @1(R4)                      ;
                          05    A4                 1F  B0 0079B          MOVW    #31, 5(R4)                                      ; 1144
                                                   79  11 0079F 91$:     BRB     100$                                            ; 0839
                                08   BE            06  D0 007A1 92$:     MOVL    #6, @LEFT_OR_RIGHT_CVT                          ; 1151
                                05   AB            68  B0 007A5          MOVW    (SRC_OR_DST), 5(SRC_OR_DST_INFO)                ; 1152
                                                   3F  11 007A9          BRB     97$                                             ; 1154
                                08   BE            06  D0 007AB 93$:     MOVL    #6, @LEFT_OR_RIGHT_CVT                          ; 1164
                          0000FFFF   8F       0C   A8  D1 007AF          CMPL    12(SRC_OR_DST), #65535                          ; 1166
                                                   23  14 007B7          BGTR    95$                                             ;
                                01            0B   A8  91 007B9          CMPB    11(SRC_OR_DST), #1                              ;
                                                   1D  12 007BD          BNEQ    95$                                             ;
                                01                 68  B1 007BF          CMPW    (SRC_OR_DST), #1                                ; 1167
                                                   18  12 007C2          BNEQ    95$                                             ;
                          0000017E   8F            56  D1 007C4          CMPL    STATE, #382                                     ; 1171
                                                   09  13 007CB          BEQL    94$                                             ;
                          0000018A   8F            56  D1 007CD          CMPL    STATE, #394                                     ;
                                                   0B  12 007D4          BNEQ    96$                                             ;
                                01            14   A8  D1 007D6 94$:     CMPL    20(SRC_OR_DST), #1                              ; 1175
                                                   05  13 007DA          BEQL    96$                                             ;
                                50                 07  CE 007DC 95$:     MNEGL   #7, STATUS                                      ;
                                                   62  11 007DF          BRB     105$                                            ;
                                6B            08   A8  90 007E1 96$:     MOVB    8(SRC_OR_DST), (SRC_OR_DST_INFO)                ; 1179
                          05    AB            0C   A8  B0 007E5          MOVW    12(SRC_OR_DST), 5(SRC_OR_DST_INFO)              ; 1180
                                              0C   AE  D5 007EA 97$:     TSTL    TURN                                            ; 1182
```

```
                                             45  12 007ED          BNEQ    103$                                   ;
                         51          0C      AC  D0 007EF          MOVL    SRC_INFO, R1                           ; 1185
                         50          04      AC  D0 007F3          MOVL    SOURCE, R0                             ;
                   01    A1          04      A0  D0 007F7          MOVL    4(R0), 1(R1)                           ;
                                             36  11 007FC 98$:     BRB     103$                                   ; 0839
                   08    BE                  06  D0 007FE 99$:     MOVL    #6, @LEFT_OR_RIGHT_CVT                 ; 1192
                                     0C      AE  D5 00802          TSTL    TURN                                   ; 1194
                                             15  12 00805          BNEQ    101$                                   ;
                         50          0C      AC  D0 00807          MOVL    SRC_INFO, R0                           ; 1197
                         51          04      AC  D0 0080B          MOVL    SOURCE, R1                             ;
         01     A0       04          A1      02  C1 0080F          ADDL3   #2, 4(R1), 1(R0)                       ;
                         05   A0     04      B1  B0 00815          MOVW    @4(R1), 5(R0)                          ; 1198
                                             18  11 0081A 100$:    BRB     103$                                   ; 1194
                         50          10      AC  D0 0081C 101$:    MOVL    DST_INFO, R0                           ; 1201
                         05   A0     08      BC  B0 00820          MOVW    @DESTINATION, 5(R0)                    ;
                                             0D  11 00825          BRB     103$                                   ; 0839
                                 00000000G   8F  DD 00827 102$:    PUSHL   #LIB$_FATERRLIB                        ; 1206
                     00000000G   00          01  FB 0082D          CALLS   #1, LIB$STOP                           ;
                                     0C      AE  D6 00834 103$:    INCL    TURN                                   ; 0748
                         03          0C      AE  D1 00837          CMPL    TURN, #3                               ;
                                             03  1A 0083B          BGTRU   104$                                   ;
                                           F7C8  31 0083D          BRW     1$                                     ;
                         50                  01  CE 00840 104$:    MNEGL   #1, STATUS                             ;
                51       34                  AE  06  C5 00843 105$: MULL3  #6, LEFT_CVT, R1                       ; 1217
                         51          30      AE  C0 00848          ADDL2   RIGHT_CVT, R1                          ;
                   14    BC          FA      A1  9E 0084C          MOVAB   -6(R1), @CVT_PATH                      ;
                                             04 00851              RET                                            ; 1219

; Routine Size:  2130 bytes,    Routine Base:  _LIB$CODE + 00F3


; 1128          1220  1
; 1129          1221  1 END                                               ! End of module LIB$$FIND_CVT_PATH.
; 1130          1222  1
; 1131          1223  0 ELUDOM




;                                    PSECT SUMMARY
;
;       Name                       Bytes                         Attributes
;
;   _LIB$CODE                         2373  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)



;                               Library Statistics
;
;                                          -------- Symbols --------         Pages        Processing
;      File                                Total     Loaded   Percent        Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1       9776        35         0          581          00:00.8
;                                      COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:LIBFINCVT/OBJ=OBJ$:LIBFINCVT MSRC$:LIBFINCVT/UPDATE=(ENH$:LIBFINCVT
;       )

; Size:          2130 code + 243 data bytes
; Run Time:         00:24.0
; Elapsed Time:     01:37.9
; Lines/CPU Min:     3057
; Lexemes/CPU-Min: 25740
; Memory Used:  433 pages
; Compilation Complete
```

0206 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBEMODH
LIS

LIBEMODD
LIS

LIBESTEMU
LIS

LIBEF
LIS

LIBEMULAT
LIS

LIBFFS
LIS

LIBFINCVT
LIS

LIBE2AREV
LIS

LIBFAO
LIS

LIBEMODG
LIS

LIBEXTV
LIS

LIBEDIV
LIS

LIBESTABL
LIS

LIBFFC
LIS

LIBEMODF
LIS

LIBEMUL
LIS

LIBFILSCA
LIS

LIBEXTZV
LIS

LIBEBCASC
LIS

LIBFAOL
LIS

0207 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBFLTUND
LIS

LIBGETSYI
LIS

LIBICHAR
LIS

LIBINITIA
LIS

LIBFIXUPF
LIS

LIBGETFOR
LIS

LIBGETINP
LIS

LIBINISHR
LIS

LIBGETDVI
LIS

LIBGETOPC
LIS

LIBFNDIMG
LIS

LIBGETMSG
LIS

LIBINDEX
LIS

LIBINSQHI
LIS

LIBGETJPI
LIS

LIBGETTAB
LIS